Password
MANAGER

UNIVERSITY OF WASHINGTON

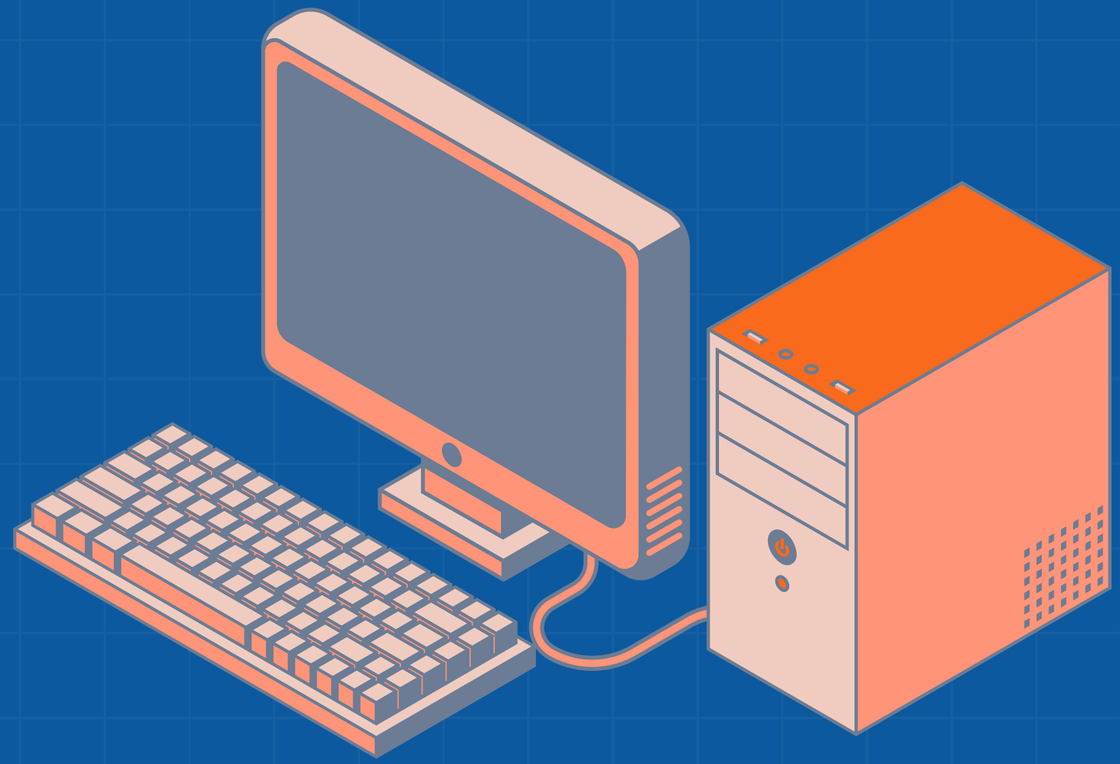


Table of Contents

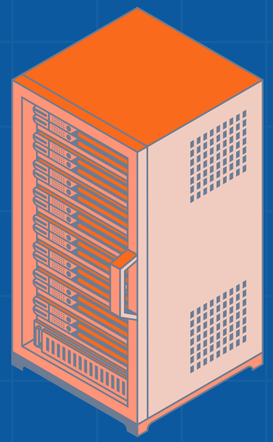
Introduction

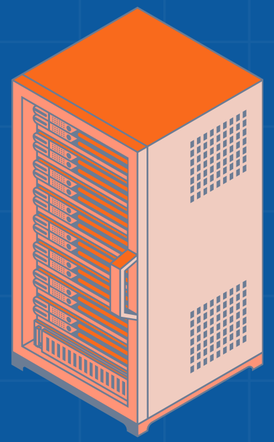
THE HUSKEY PASSWORD MANAGER IS A WEB-BASED APPLICATION USED TO STORE AND MANAGE PASSWORDS.

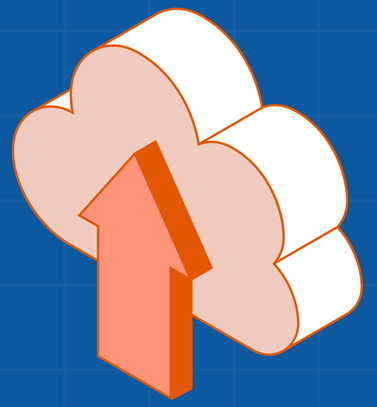
USERS CAN LOG IN WITH A SET OF CREDENTIALS TO VIEW AND MANAGE DEPARTMENTAL PASSWORD VAULTS.

IT OPERATES ON A DOCKER-BASED ENVIRONMENT WITH THREE KEY COMPONENTS.

Web Application **COMPONENTS**







BROWSER

NGINX SERVER

REVERSE PROXY FOR CLIENT REQUESTS, FORWARDING THEM TO BACKEND SERVERS FOR PROCESSING.

HANDLES STATIC CONTENT DELIVERY OF HTML, CSS, AND JAVASCRIPT FILES.

PHP SERVER

A SERVER-SIDE SCRIPTING LANGUAGE THAT EXECUTES WEB APPLICATION FUNCTIONS THAT REQUIRE SERVER-SIDE PROCESSING AND DATABASE INTERACTION.

SUCH AS USER AUTHENTICATION, VAULT OPERATIONS AND MANAGEMENT, PAGE RENDERING, AND BACKEND LOGIC AND PROCESSING.

MYSQL DATABASE

STORES USER CREDENTIALS AND PASSWORD VAULT DATA.

Insecurities

UNENCRYPTED CONNECTIONS

HTTPS NOT SETUP IN NGINX

NONEXISTENT LOGGING PRACTICES

BROKEN AUTHENTICATION AND ACCESS CONTROLS

NO SESSION MANAGEMENT BY PHP SERVER

MISCONFIGURATIONS

USE OF HTTP AND INSUFFICIENT COOKIE MANAGEMENT

SQL VULNERABILITIES

UNPARAMETERIZED QUERIES

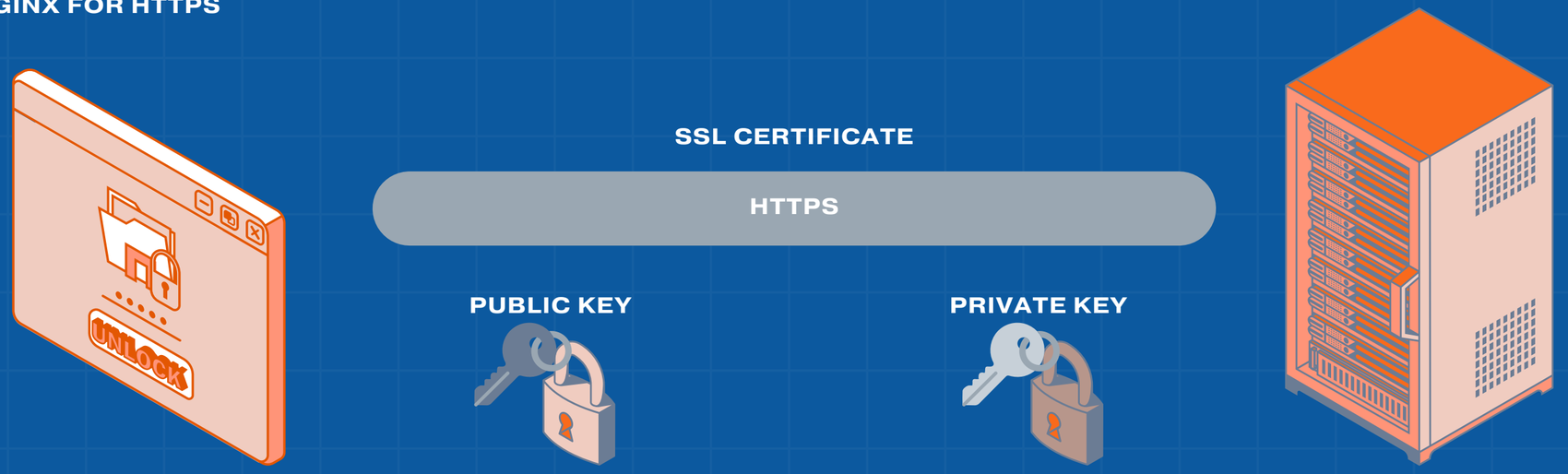
XSS VULNERABILITIES

UNSANITIZED USER INPUTS

Cryptography

ENCRYPT DATA BETWEEN THE CLIENT AND SERVER

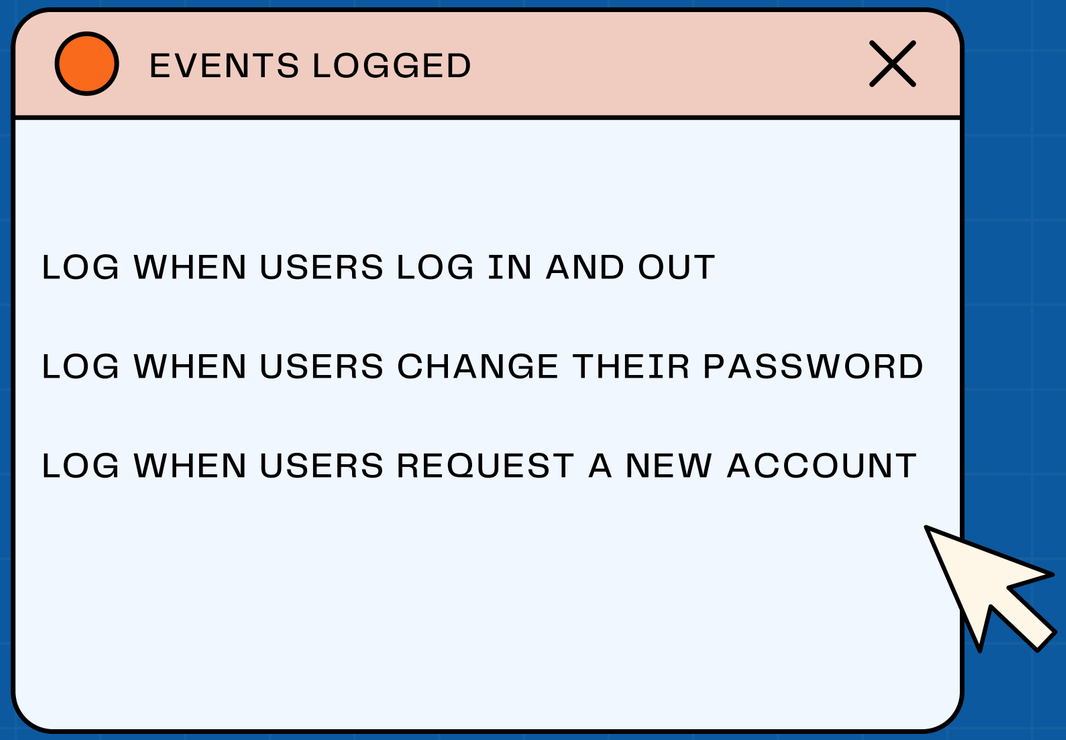
1. CONFIGURE OPENSSL
2. GENERATE A PUBLIC-PRIVATE KEY PAIR
3. CREATE A CERTIFICATE SIGNING REQUEST (CSR)
4. SIGN THE CSR WITH A CERTIFICATE AUTHORITY (CA)
5. GENERATE AND INSTALL THE SSL CERTIFICATE
6. CONFIGURE NGINX FOR HTTPS



Logging

PROTECT ASSETS BY RECORDING SYSTEM ACTIVITIES

1. CREATE FREE LOGGLY ACCOUNT
2. COPY LOGGLY TOKEN INTO LOGGLY-LOGGER.PHP
3. ADD LOGGLY COMPONENT TO LOGIN.PHP
4. ADD LOGGER CODE TO APPROPRIATE CODE
(E.G. ELSE STATEMENT IN LOGIN CODE)
6. VERIFY LOGGER FUNCTIONALITY



Authentication and **AUTHORIZATION**

PRESENTATION TIER

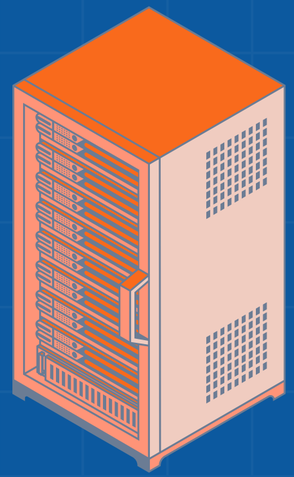
USERNAME/
PASSWORD LOGIN

LOGIC TIER

VERIFY
CREDENTIALS

AUTHENTICATION
PASS

SESSION TOKEN

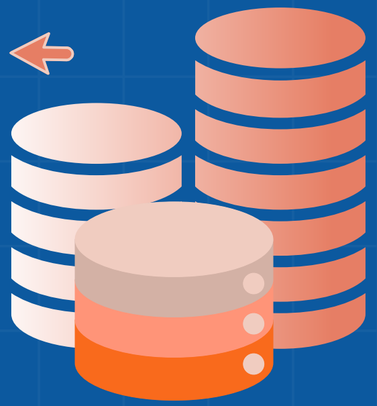


AUTHORIZATION
GRANTED

DATA TIER

VERIFY ROLE BASED
ACCESS CONTROL

ADMIN
DEV
EXEC
HR



Application **SECURITY**

XSS

IMPACTS:

- SESSION HIJACKING
- DATA THEFT
- SOCIAL ENGINEERING

MITIGATION:

- INPUT SANITIZATION
- OUTPUT ESCAPING
- FRAMEWORK-BASED PROTECTIONS

SQL INJECTIONS

IMPACTS:

- DATA BREACH AND MANIPULATION
- UNATHORIZED ACCESS
- DATABASE OVERLOAD (DOS)
- LOSS OF INTEGRITY

MITIGATION:

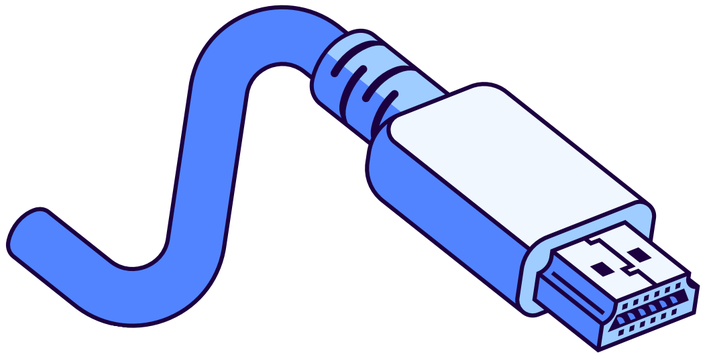
- PARAMETERIZED QUERIES
- INPUT VALIDATION

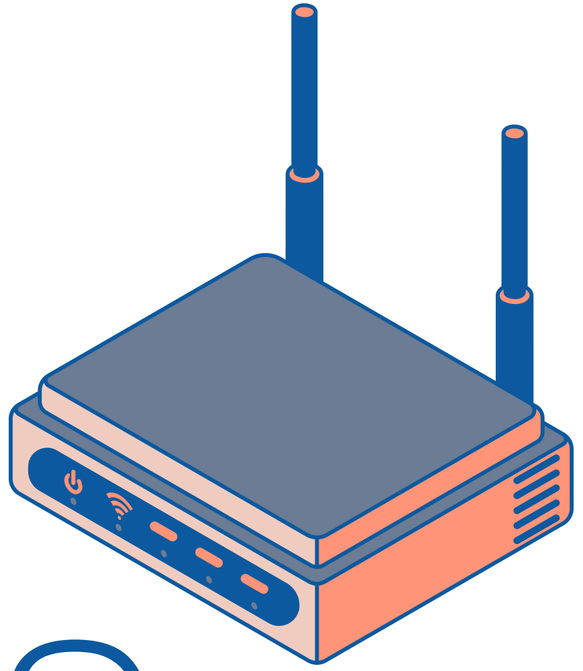
Hardening

SECURITY ISSUES IDENTIFIED VIA ZAP		REMEDIATION
CWE-613	→ INSUFFICIENT SESSION EXPIRATION	→ ADD SESSION_START() TO LOGOUT.PHP
CWE-80	→ IMPROPER NEUTRALIZATION OF SCRIPT-RELATED HTML TAGS (XSS)	→ ESCAPE INPUTS AND CONVERT TO PLAINTEXT ADD HTMLSPECIALCHARS() TO PHP “ECHO” STATEMENTS
CWE-284	→ IMPROPER ACCESS CONTROL	→ REMOVE PORTS “3306:3306” FROM DOCKER-COMPOSE.YAML
CWE-89	→ IMPROPER NEUTRALIZATION OF SPECIAL ELEMENTS USED IN A SQL COMMAND	→ CONVERT USER INPUTS TO PLAINTEXT ADD PREPARED STATEMENTS TO LOGIN.PHP

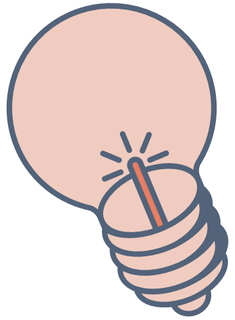
MONITORING AND MAINTAINING SECURITY

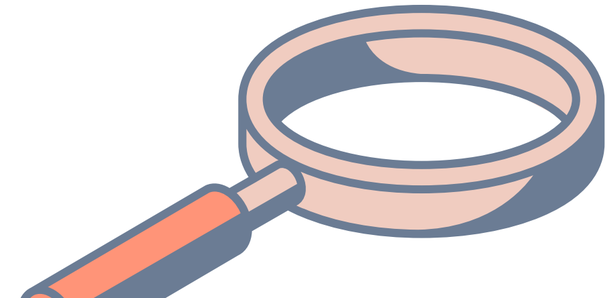
- CONTINUOUS MONITORING OF LOGS
- REGULARLY RUN SCANS IN ZAP
- REGULARLY REVIEW AND UPDATE THIRD-PARTY LIBRARIES AND FRAMEWORKS
- APPLY UPDATES AND SECURITY PATCHES FOR THE SERVER AND DATABASE
- CONDUCT PERIODIC PENETRATION TESTS TO SIMULATE REAL-WORLD ATTACKS AND UNCOVER HIDDEN VULNERABILITIES
- CONDUCT CODE REVIEWS FOCUSING ON INPUT VALIDATION, OUTPUT ENCODING, AND DATABASE QUERY SANITIZATION
- REGULARLY AUDIT ROLES AND PERMISSIONS TO ENSURE USERS HAVE THE MINIMUM NECESSARY ACCESS (PRINCIPLE OF LEAST PRIVILEGE)
- IMPLEMENT MULTI-FACTOR AUTHENTICATION (MFA)





Questions



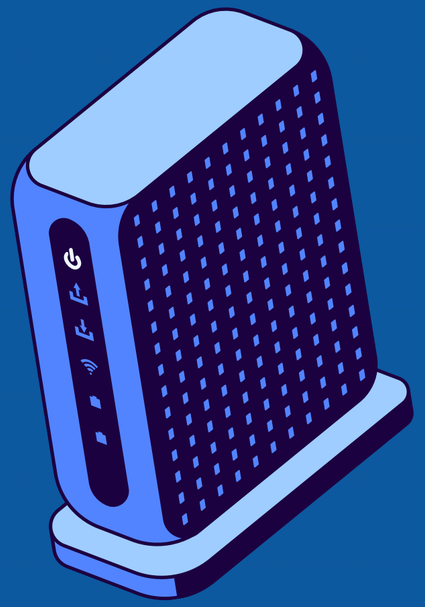


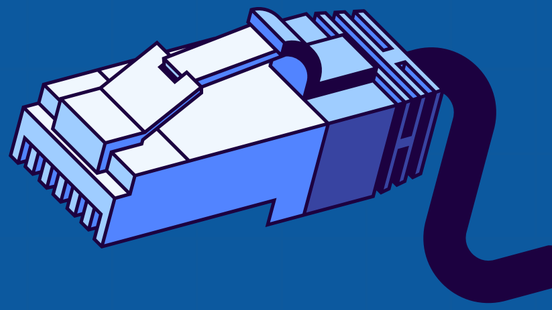
REFERENCES

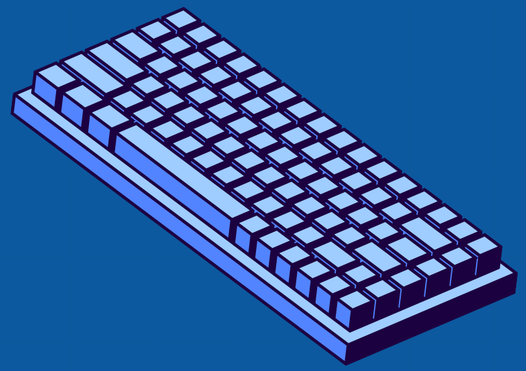
DOCKER HUB. (N.D.). HUB.DOCKER.COM. HTTPS://HUB.DOCKER.COM/_/NGINX

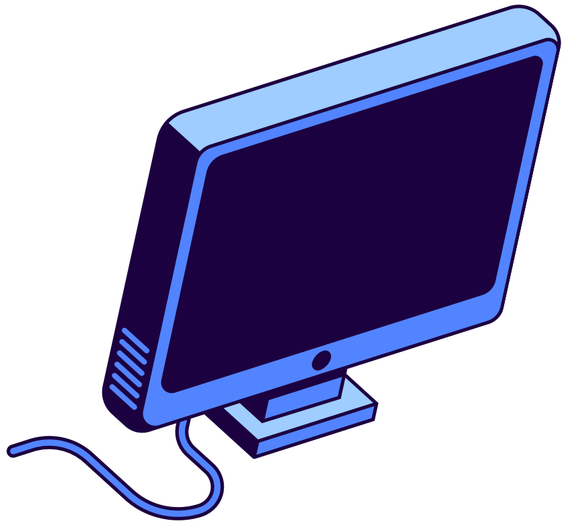
THAKUR, A. (2018, NOVEMBER 17). OPTIMIZING PHP-FPM FOR HIGH PERFORMANCE. GEEKFLARE. HTTPS://GEEKFLARE.COM/PHP-FPM-OPTIMIZATION/

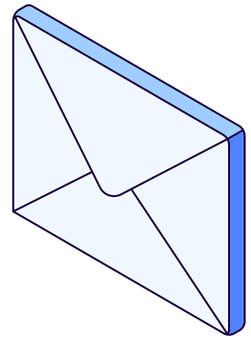
ANON. (2024). GITHUB.COM. HTTPS://GITHUB.COM/UW-AREIFERS/AUT-24-UW-CYBERSEC-HUSKEY-MANAGER/TREE/W4-LOGGING











Thank you